UNITED STATES PATENT APPLICATION

FOR

METHOD AND SYSTEM FOR DECODING BIPHASE-MARK ENCODED DATA

Inventor(s):

Zoltan TOTH
Kenneth D. SMITH, Jr.
Hung B. VO

Sawyer Law Group LLP
2465 E. Bayshore Road
Suite 406
Palo Alto, CA 94303

METHOD AND SYSTEM FOR DECODING BIPHASE-MARK ENCODED DATA

FIELD OF THE INVENTION

[001] The present invention relates to decoding a digital audio data stream, and more particularly to decoding a biphase-mark encoded data stream in the digital domain using a software-based frequency locked loop.

BACKGROUND OF THE INVENTION

[002] Digital audio receivers that receive and decode digital audio data are used in a variety of consumer and professional audio products such as audio/video receivers, DVD players and recorders, and personal video recorders, for example. Several audio interfaces exist to allow different devices to communicate digital audio data with one another. The worldwide audio standard is referred to the as International Electrotechnical Commission 60958 (IEC-60958). Two related standards include the Audio Engineering Society's AES3 standard, and the Sony/Philips Digital Interface (S/PDIF). One commonality among these standards is that they modulate the signal using biphase-mark encoding (or simply biphase encoding) to transmit digital audio information in serial format.

[003] Biphase-mark encoded standards require that the digital audio data be transmitted in blocks, where each block comprises 192 consecutive frames for each of two channels. Figure 1 is a diagram illustrating the IEC958 frame format. A frame 10 comprises two sub-frames 12, one for each channel. The rate of

transmission of the frames corresponds to the source sampling frequency. Samples of the audio signal are transmitted in two channels, and a receiver is able to extract the clock from the transmitted signal. Each sub-frame 12 is divided into thirty-two time slots for storing data bits. The first four bits form a preamble, the following four bits include auxiliary data, the following twenty bits contain an audio sample, and the last four bits contain control information.

[004] To facilitate clock recovery from the data stream and to minimize the DC component on the transmission line, time slots 4 to 31 are encoded in biphase-mark. Each time slot, which transmits one bit of data, always starts with a transition or edge of the signal and ends with a transition of the signal. Further, each time slot is represented by two consecutive binary states, where the occurrence of a transition in the middle of a given time slot indicates that a "1" is encoded, while the absence of a transition in a time slot indicates that a "0" is encoded.

[005] Figure 2 illustrates an example of biphase-mark encoding. A clock signal 16 is shown for reference. An example set of source data 18 is shown that is to be encoded, followed by example encoded data 20 after biphase-mark encoding. The example clock rate is twice the data transmission rate. Each bit from the source data 18 is transmitted in the encoded data 20 in a time slot of length T. As shown, "1s" in the source data 18 are encoded in the encoded data 20 with transitions occurring at .5T, while the "0s" have no transition.

[006] Referring again to Figure 1, the preambles 14 of each sub-frame 12 are specific patterns that provide synchronization and identification of sub-frames 12 and blocks. This allows the receiver to lock onto the data sample within one sub-frame 12.

[007] Figure 3 is a block diagram illustrating the three preambles used to identify the beginning of the sub-frames and blocks; and Figure 4 shows relative waveforms for the preambles. There are three defined preambles 14: one for each channel and one to indicate the beginning of a channel and a block. The corresponding channel coding for each of the preambles 14 is also shown. Preamble "B" represents the beginning of channel A and a block. Preamble "M" represents the beginning of channel A only. And preamble "W" represents the beginning of channel B. In broadcasting environments, the letters B, M, and W are denoted by Z, X, and Y, respectively.

[008] So that they can be easily identified by the receiver, each preamble 22 contains biphase coding violations, as shown in Figure 4. Biphase-mark data is required to transition at every time slot, but each preamble 22 violates that requirement twice because each preamble 22 has two time slots, or bit boundaries, with no transitions.

[009] Biphase-mark encoding has the advantage of providing a self-clocking data signal. Conventional digital audio receivers typically include an analog phase-locked loop (PLL) to recover the clock signal from the data signal. The PLL includes a digital edge-triggered phase detector coupled to a voltage-

controlled oscillator. Through a feedback loop, the voltage-controlled oscillator is used to generate an independent clock signal having a frequency designed to match the clocking of incoming biphase-mark encoded data signal.

[010] Although conventional digital audio receivers work for their intended purposes, the sampling frequency of incoming digital audio data may change significantly. In addition, the signal may suffer from jitter and noise, making it even more difficult for the digital audio receiver to decode the audio data. Because the PLL's used in conventional digital audio receivers are analog, these receivers take longer to adapt to the changing input sampling frequency due to low pass filter characteristics. If the digital audio receiver is incapable of keeping up with the changing sampling frequency of the input signal, then unsatisfactory decoding of the digital audio signal will result.

[011] Accordingly, what is needed is an improved method and system for decoding biphase encoded data. The present invention addresses such a need.

SUMMARY OF THE INVENTION

[012] The present invention provides a method and system for decoding a biphase-mark encoded input stream. Aspects of the present invention include receiving an external biphase-mark input stream by a receiver module; recovering timing information from the input stream; decoding the input stream to generate decoded data and storing the decoded data in a data buffer; reading, by an audio out module, the decoded data from the data buffer at a rate determined by a programmable clock; using the timing information from the receiver module

to calculate a sampling frequency of the input stream; and adjusting the frequency of the programmable clock to substantially match the sampling frequency so that the audio out module reads the decoded data from the buffer at substantially the same rate that the receiver module inputs the decoded data into the data buffer.

[013] According to the method and system disclosed herein, the present invention decodes the biphase input stream entirely in the digital domain using a software-based frequency locked loop, which matches the frequency of the programmable clock with the sampling frequency of the input stream in order to synchronize the audio out module with the receiver module.

BRIEF DESCRIPTION OF THE DRAWINGS

[014] Figure 1 is a diagram illustrating the IEC-60958 frame format.

[015] Figure 2 illustrates an example of biphase-mark encoding.

[016] Figure 3 is a block diagram illustrating the three preambles used to identify the beginning of the sub-frames and blocks.

[017] Figure 4 shows relative waveforms for the preambles.

[018] Figure 5 is a diagram illustrating an integrated biphase decoding system in accordance with one preferred embodiment of the present invention.

[019] Figure 6 is a diagram illustrating components and process flow for the 1T cycle detector and the Jitter Filter Logic module.

[020] Figure 7 is a diagram illustrating waveforms of the "X" preamble.

DETAILED DESCRIPTION OF THE INVENTION

[021] The present invention relates to a method and system for decoding biphase encoded data. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiments and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features described herein.

[022] The present invention provides a biphase-mark decoding system. The system includes a digital audio receiver that may be integrated with an audio video decoder for decoding a biphase-mark input stream in the digital domain and without the use of an analog PLL.

[023] Figure 5 is a diagram illustrating an integrated biphase-mark decoding system in accordance with one preferred embodiment of the present invention. The biphase-mark decoding system 100 includes a digital audio receiver module 102 coupled to a host processor 104. The host processor 104 is shared between audio and video chips (not shown), and executes system control software 120.

The system control software 120 performs a variety of functions for controlling the digital audio receiver module 102, including implementation of the frequency locked loop (FLL).

[024] Input to the receiver module 102 includes a digital audio data stream 112, which is biphase-mark encoded, preferably using the IEC60958, AES3, or S/PDIF standards. The input stream 112 is typically input to the system 100 through an optical or coaxial 75-ohm cable from a device, such as a DVD or CD player, for example.

[025] Also input to the receiver module 102 for timing control are two clocks: a system clock (SYSCLK) 114, and an audio clock (XCK) 116. The system clock 114 is an on chip clock and is not programmable by the host processor 104. The audio clock 116 is also an on chip clock and it is implemented as a programmable clock oscillator. In a preferred embodiment, the system clock 114 is capable of significantly higher frequencies than the audio clock 116. For example, the system clock 114 may reach frequencies of approximately 148 MHz, while the audio clock 116 may only reach frequencies up to approximately 38 MHz.

[026] According to the present invention, the receiver module 102 uses either the system clock 114 or the audio clock 116 to recover timing information from the input data stream 112. In a preferred embodiment, the system clock 114 is used for decoding data because it provides the highest precision, and the audio clock 116is only used for rate control by the audio clock register 141, as

described below. The decoder 126 then uses a portion of that timing information to decode the data stream 112, and the decoder 126 stores the decoded audio data 118 in a data buffer 106. The audio out module 108 then reads the decoded data 118 from the data buffer 106 at a rate determined by the audio clock 116, and outputs the data in framed serial format to a digital-to-analog converter (DAC) 110 for subsequent delivery to a device, such as an amplifier and a speaker.

[027] The system control software 120 uses the timing information gathered by the receiver module to calculate the sampling frequency of the input stream 112 and then adjusts (increases or decreases) the frequency of the audio clock 116 to substantially match the input sampling frequency. By using a software-based FLL to cancel any error between the audio clock 116 and the sampling frequency of the input stream, the present invention ensures that the audio out module 108 reads the decoded audio data 118 from the buffer 106 at substantially the same rate that the receiver module 102 inputs the audio data 18 into the data buffer 106.

[028] According to the present invention, the software-based frequency locked loop implemented by the system control software 120 provides a method for adjusting characteristics of the receiver module 102 to obtain the best trade-off between jitter and noise tolerance, as explained further below.

[029] The receiver module 102 includes a 1T cycle detector 122, jitter filter logic 124, a decoder 126,CONFIGURATION registers 128, a counter module 130 and

a counter interrupt 132. The receiver module 102 functions according to values of one or more of the CONFIGURATIONregisters 128, which are set by the system control software 120.

[030] In operation, the 1T cycle detector 122 and the jitter filter logic 124 are used to first establish 1T timing of the input stream 112. The system control software 120 in conjunction with the counter module 130 then establish the sampling frequency of the input stream 112 using either the system clock 114 or the audio clock 116. Based on this timing information, the decoder 126 can decode the input stream 112 and fills the buffer 106 at a rate corresponding to the rate of the input stream 112. The audio out module 108 then reads the data from the data buffer 106 at a rate set by the audio clock 116.

[031] As the input stream 112 is received, the receiver module 102 uses either the system clock 114 or the audio clock 116 to recover timing information from the input stream 112. The receiver module 102 feeds the timing information back to the system control software 120 as a frequency count. The system control software 120 compares this frequency count with an expected value and calculates the difference. The system control software 120 then adjusts (increases or decreases) the frequency of the audio clock 116 according to this difference via an output clock frequency control register (fxck) 132 to substantially match the sampling frequency of the input and the output.

[032] As explained further below, the timing information reported to the system control software 120 includes a count of the number of clock cycles, and a count

of the number of sub-frames detected in the input signal. These two counts are used by the system control software 122 to determine the sampling frequency of the input stream 112.

[033] Although the system clock 114 is not programmable, its frequency is much higher than that of the audio clock 116 and therefore it is used to determine the sampling frequency of the input stream 112 at system startup when the frequency of the input stream 112 is still unknown, or when the input sampling frequency is very high and the system 10 cannot keep up because the audio clock 116 has too few clock cycles within the measurement interval. Due to its higher frequency, the system clock 114 provides a higher resolution than the audio clock 116, and therefore results in a more accurate measurement.

[034] When the system 10 starts, the system control software 120 makes an assumption about the sampling frequency of the input stream 112. For example, a default sampling frequency of 48 kHz may be assumed. Once timing has been established, the system control software 120 may transition from using the system clock 114 to using the audio clock 116 to continue establishing timing of the input stream 112.

[035] It should be noted that when the system clock 114 is used by the system control software 120 to implement the FLL, the FLL forms an open loop because the frequency of the system clock 114 cannot be changed by the system control software 120. However, when the audio clock 116 is used, the FLL forms a closed loop because the frequency of the audio clock 116 is adjustable by the

system control software 120 to reduce the error between the audio clock 116 and the calculated sampling frequency of the input stream 112.

[036] The operation of the decoding system 10 will now be explained in further detail. The receiver module 102 begins by converting the asynchronous IEC60958 input stream 112 into a synchronous input stream 112 for internal use. Processing of the input stream 112 begins with recovering the clock by sensing the frame and block synchronization patterns, which identify the location of each channel. This is referred to as 1T timing information. Given the asynchronous IEC60958 input stream 112, the 1T cycle detector 122, which is controlled by the jitter filter logic 124, measures the duration of the 1T interval (also called unit interval) in the input stream 112. When the 1T cycle detector 122 completes detecting the 1T timing in the input stream 112, the incoming data can be accurately sampled and the audio clock 116 should be synchronized with the timing in the input stream 112.

[037] The input stream 112 inherently includes jitter and noise. Jitter and noise affect the rising and falling edges of the signal, making clock recovery difficult because the slopes of the edges in each timing interval may be different.

[038] According to the present invention, the jitter filter logic 124 and the 1T cycle detector 122 accurately recover the clock through a combination of the following features:

1) adaptive jitter tolerance;

2) extracting the 1T timing of the input stream 112 only from the "M" or "X"

preamble patterns 11100010 or 00011101; and

3) by measuring the timing intervals by measuring the time between pairs of consecutive edges of the same type in the signal, rather than measuring the time between pairs of consecutive rising and falling edges, as explained below.

[039] ADAPTIVE JITTER TOLERANCE

[040] The adaptive jitter tolerance of the jitter filter logic 124 provides the 1T cycle detector 122 with an adjustable jitter window for accepting clock edges that arrive early or late. This allows the 1T cycle detector 122 to adapt to the sampling frequency of the input stream 112, which can vary from 32 kHz to 192 kHz. In a preferred embodiment, the jitter window is established by measuring 1T timing as a predefined number of system clock cycles, such as 15 cycles for example. The system control software 120 then divides the number of clock cycles by any number divisible by two to provide a jitter window that is proportional to the input sampling frequency. Using a number divisible by 2 simplifies the implementation because the division can be carried out by bitwise shifting the number of clock cycles towards the least significant bity. Such a method of division results in a logarithmic set of steps which is optimal for cost and performance.

[041] For example, dividing the number of cycles by 2 provides a jitter window that is 50 percent of the sampling frequency. Preferably, an optimal jitter window may be created by dividing the number of cycles by 8 to provide a jitter window

that is 12.5% of the sampling frequency. For a sampling frequency of time T, this means that a legal edge transition is recognized anytime within T/8.

[042] The jitter window is defined by determining a lower and upper limit of the 1T interval. In a preferred embodiment, the lower and upper limit of the 1T interval is based on a value of a jitter control register (not shown) in the CONFIGURATIONregisters 128 set by the system control software 120.

[043] 1T TIMING

[044] Given the lower and upper limit from the jitter filter logic 124, the 1T cycle detector 122 processes the input stream 112. The 1T cycle detector 122 decides the 1T interval that satisfies the jitter requirement. There are two windows in the input stream 112 that are monitored; a preamble window in which a preamble pattern should appear, and a data window in which data should appear. In the data window, the duration of each signal transition from high-to-low or low-to-high is either 1T or 2T. In the pre-amble window, a 3T transition happens with a unique pattern.

[045] According the present invention, the 1T cycle detector 122 extracts the 1T timing of the input stream 112 only from the "M" or "X" preamble pattern (11100010 or 00011101). The 1T cycle detector 122 accomplishes this by looking for two consecutive 3T's, or 6T, followed by a 2T pattern. The search method is to look for the 6-to-2 timing ratio in this sequence, which can only happen during the 3T-3T-1T-1T pattern of the "M" or "X" preamble. Once this pattern is found, the 2T value is divided by 2 to yield the 1T value. In an

alternative embodiment, once the 8T pattern of 3T-3T-1T-1T is found, the 8T value could be divided by 8 to yield the 1T value.

[046] Every time a 6T-2T sequence appears, the 1T interval value is updated. The 1T interval value may be updated either by overwriting the previous 1T value or by averaging the current and previous 1T values. Based on the stored 1T value, a signal is sent to the decoder module 126 which is sampled in the middle of every 1T interval detected in the input stream 112.

[047] Figure 6 is a diagram illustrating components and process flow for the 1T cycle detector 122. The internal architecture of 1T cycle detector 122 includes a counter 200, a 3-times multiplier 204, a jitter controller 206, a decision block 208, and a divide by two & latch block 202. An explanation of these functional blocks will follow the data path of the signal.

[048] The 1T cycle detector 122 accepts two inputs, the system clock 114 and the input stream 112, and generates "1T value" as its output. The 1T cycle detector 122 tries to extract the "1T value" from the detected "X" or "M" preamble pattern, 11100010 or 00011101.

[049] The counter 200 measures the time interval between edge transitions in the input stream 112 by counting system clock cycles 114. However, instead of measuring pairs of adjacent rising and falling edges, the counter 200 counts the number of system clock 114 cycles occurring between transitions of adjacent edges of the same type (both rising or both falling). This is done to prevent any differences between rise and fall times from translating into input jitter. Receivers

often employ hysteresis to reduce their susceptibility to input noise. Even ordinary receivers exhibit some form of hysteresis. Unfortunately, hysteresis makes the receiver susceptible to differences in the rise and fall times of the signal. Consequently, measuring timing between consecutive rising and falling edges may increase the jitter of the signal as seen by the receiver, rendering it less tolerant to jitter originating at the source. Counting the number of system clock 114 cycles occurring between transitions of adjacent edges of the same type in accordance with the present invention eliminates this error.

[050] The output of the counter 200 are an old count value 210 and a new count value 212. Figure 7 is a diagram illustrating waveforms of the "X" preamble. As shown, the old count value 210 represents the number of system clock 114 cycles occurring during the 6T (3T + 3T) interval of the "X" preamble. The new count value 212 represents the number of system clock 114 cycles occurring during the subsequent 2T (1T + 1T) interval.

[051] The 3-time multiplier 204 multiplies the new count value 212 by 3, and produces the result 3 times of new count value 214. The jitter controller 206 retrieves the jitter window setting (e.g. +/-12.5%) from a register that is set by the system control software 120 while the input stream is being processed. The jitter controller 206 adds the jitter window setting to the 3-times of the new count value 216 in order to produce the "upper limit" value 216 of the jitter window. The jitter controller 206 subtracts the uncertainty value from the 3-times of the new count value 214 in order to produce the "lower limit."

[052] In the decision block 208, the old count value 210 is compared with the upper limit 218 and lower limit 216. If the old count value 212 is between the upper limit 218 and the lower limit 216, a "match" signal 220 is generated by the decision block 208. This match signal 220 indicates that a timing sequence of 3 time units followed by 1 time unit was found, which is only possible in the "M" or "X" preamble (3T-3T-1T-1T) according to the IEC60958 standard.

[053] The divide-by-2 & latch 202 waits for the match signal 220. When the divide-by-2 & latch 202 detects the match signal 220, the new count value 212 is divided by 2 and latched into the output register as the 1T value 224.

[054] According to a further aspect of the present invention, the 1T cycle detector 122 also reports the occurrence of a bad input stream 112 to the system control software 120. In a preferred embodiment, the 1T cycle detector 122 notifies the system control software of a bad signal using a bad signal interrupt 136. The bad signal Interrupt 136 is generated if any of the following conditions occur:

edge-to-edge interval of 4T or longer is detected,

a 3T edge-to-edge interval appears in a data window,

a preamble is not detected when expected, and

a preamble is detected at a wrong position.

[055] DECODER 126

[056] Referring again to Figure 5, while the 1T cycle detector 122 is processing the input stream 112 and updating the 1T value 224, the decoder module 126

decodes the input stream 112 using the 1T value 224. The decoder module 126 includes shift registers (not shown) for storing the polarity of the input stream 112 when the 1T interval lapses. The shift registers are of sufficient size to store the preamble pattern. The data shift registers are constantly compared with the patterns of the B, M, and W preambles to identify the start of sub-frames. When the decoder module 126 detects a preamble, a preamble start signal is sent to the counter module 130.

[057] COUNTER MODULE 130

[058] The function of the counter module 130 is to report the frequency of the input stream 112 to the system control software 120. According to the present invention, the counter module 130 uses either the system clock 114 or the audio clock 116 in order to calculate the input stream 112 frequency, as instructed by the system control software 120. The counter module 130 uses a system clock counter register 140, an audio clock register 141, and a sub-frame counter register 142, respectively, to count the system clock 114 or the audio clock 116 cycles while simultaneously counting the sub-frames from the input stream 112.

[059] One feature of the present invention is that it maximizes the utilization of the system clock counter register 140 or the audio clock counter register 141 because the more bits used in the counter 140, the more precise the timing measurement will be. The count is maximized by maintaining the counting for as many frames as possible. Using the number of clock cycles counted during past frames, it determines when to stop counting by adding that count to the

current value of the system clock counter register 140 or the audio clock counter register 141 at the beginning of each frame to determine if the counter 140 will overflow or not by the end of the frame. If the counter register 140 or 141 will not overflow, then counting continues. Otherwise, the counter register 140 or 141 is deemed full and counting ends.

[060] When the clock counter register 140 or 141 becomes full, the values of the clock counter register 140 or 141 and the sub-frame counter register 142 are reported to the system control software 120. The counter module 130 also compares the values of the clock counter register 140 or 141 and the sub-frame counter register 142 with expected values indicated by the system control software 120. If the differences pass a predetermined threshold, then the counter module 130 generates a counter interrupt 132 to notify the system control software 120 of a change in the input stream 112 sampling frequency.

[061] INPUT STREAM FREQUENCY CALCULATION

[062] The system control software 120 calculates the frame or sample frequency of the input stream 112 using the values of the clock counter register 140 or 141 and sub-frame counter register 142 and the frequency of whichever clock, the system clock 114 or audio clock 116 that was used to perform the counting. More specifically, the sample frequency is determined by:

[063] Sample Freq.= (sub-frame counter value / (2 * clock counter value)) * clock frequency

[064] Based on the calculated sample frequency, the system control software 120 uses the clock frequency control (fxck) signal 134 to speed up or slow down the audio clock 116.

[065] A method and system for decoding a biphase-mark signal has been disclosed that uses a software controlled frequency locked loop to synchronize an audio out module with a receiver module. The present invention has been described in accordance with the embodiments shown, and one of ordinary skill in the art will readily recognize that there could be variations to the embodiments, and any variations would be within the spirit and scope of the present invention. Accordingly, many modifications may be made by one of ordinary skill in the art without departing from the spirit and scope of the appended claims.